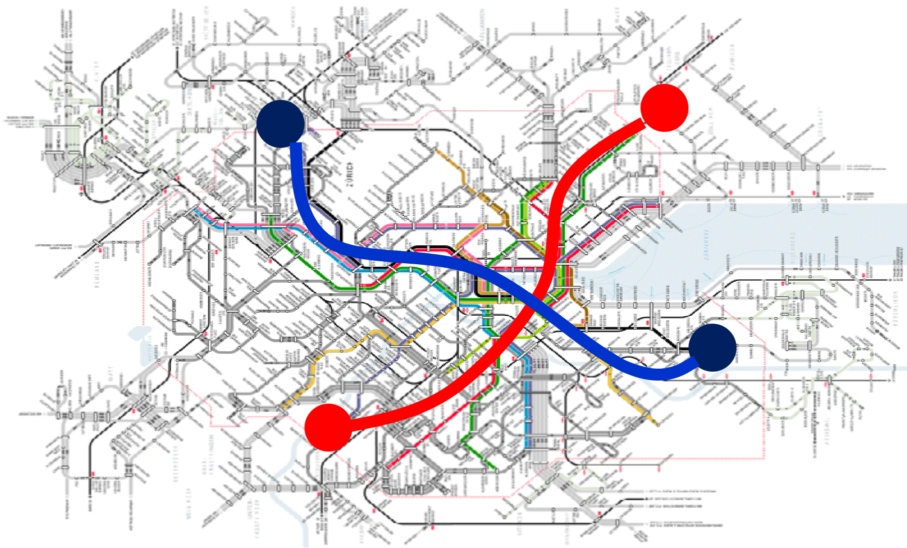
Markov Approximation for Combinatorial Network Optimization

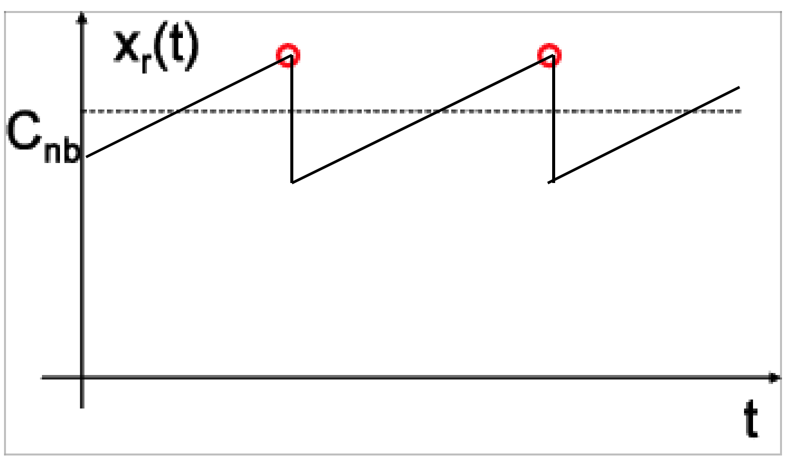
INFOCOM 2010 – TS37

Minghua Chen
Department of Information Engineering
香港中文大學
The Chinese University of Hong Kong

Resource Allocation is Critical



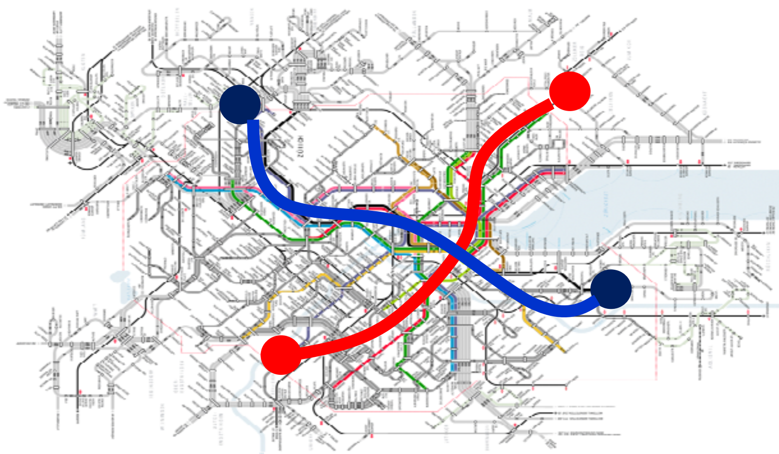
- Utilize resource
  - Efficiently
  - Fairly
  - Distributedly



- TCP: A bottom-up example
  - No loss: increase the rate
  - Loss detected: decrease the rate

Convex Network Optimization: Popular and Effective

- Formulate resource allocation as a utility maximization problem [Kelly 98, Low et. al. 99, …]



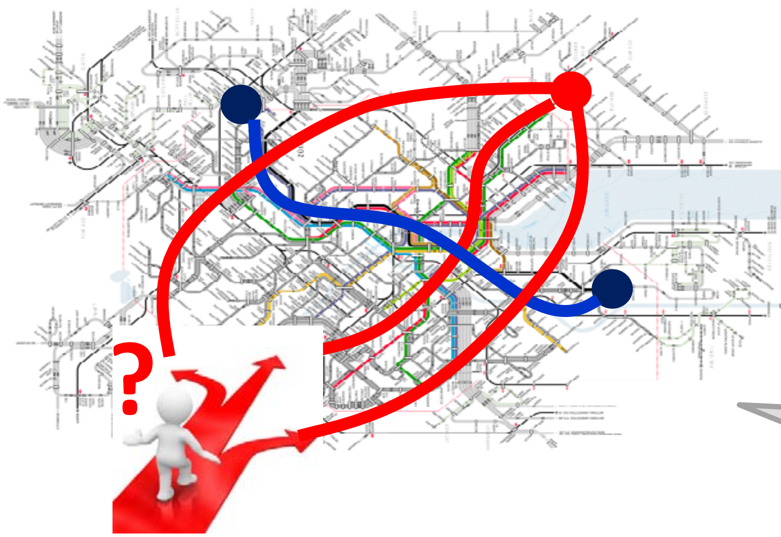
$$\max_{\boldsymbol{x} \geq 0} \quad \sum_{s \in S} U_s(x_s)$$

$$\text{s.t.} \quad Ax \leq C$$

- Design distributed solutions
  - Local decision, adapt to dynamics

Combinatorial Network Optimization: Popular but Hard

□ Joint routing and flow control problem



$$\max_{\boldsymbol{x}\geq 0, A} \quad \sum_{s\in S} U_s(x_s)$$
$$\text{s.t.} \quad Ax \leq C,$$
$$A \in \{\text{feasible routing matrices}\}$$

Combinatorial

□ Many others: Wireless utility maximization, channel assignment, topology control …

Observations and Messages

Convex: solved	Combinatorial: open

• Top-down approach
  • (mostly) convex problems
 	
  • Theory-guided distributed solutions
	• Top-down approach
	  • Combinatorial problems
	
	  • ??

□ **This paper:** Theory-guided design for distributed solutions for combinatorial network problems

Markov Approximation: Our New Perspective

Convex network problems

↓

Formulation

↓

Penalty/decomposition approach

↓

Primal/dual/primal-dual design

Combinatorial network problems

↓

Formulation

↓

Log-sum-exp **approximation**

↓

Distributed Monte Carlo **Markov** Chain

Generic Form of Combinatorial Network Optimization Problem

$$\max_{f \in \mathcal{F}} \quad W_f.$$

- System settings:
  - A set of user configurations, $f = [f_1, f_2,$ $\in \mathcal{F}$
  - System performance under f, W_f

Exponentially large

- Goal: maximize network-performance by choosing configurations

NP-hard

Examples

- Wireless network utility maximization
  - Configuration f: independent set
- Path selection and flow control
  - Configuration f: one combination of selected paths
- Channel assignments in WiFi networks
  - Configuration f: one combination of channel assignments
- Local balancing in distributed systems…

New perspective

New perspective and new solutions

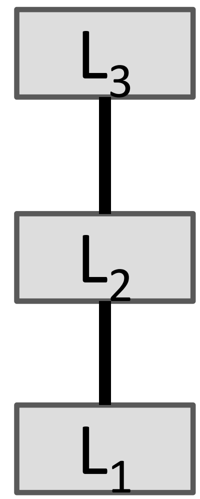
Wireless Network Utility Maximization

$$\max_{\boldsymbol{z}\geq 0, \boldsymbol{p}\geq 0} \sum_{s\in S} U_s(z_s)$$

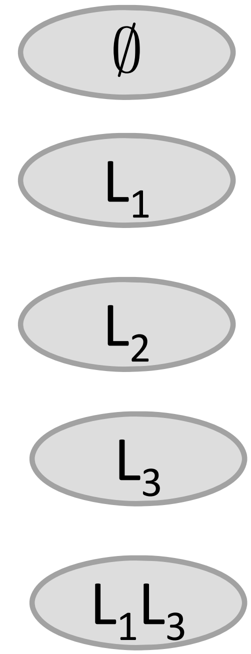
Wireless link capacity constraints

$$\text{s.t.} \quad \sum_{s:l\in s, s\in S} z_s \leq \sum_{f:l\in f} p_f, \quad \forall l \in L$$

$$\sum_{f\in \mathcal{F}} p_f = 1$$



3-links interference graph



independent sets

- z_s: rate of user s
- L: set of links, each with unit capacity
- $\mathcal{F}$: set of all independent sets (configurations)
- p_f: percentage of time f is active

Scheduling Problem: Key Challenge

$$\min_{\boldsymbol{\lambda}\geq 0} \quad \max_{\boldsymbol{z}\geq 0} \quad \sum_{s\in S} U_s(z_s) - \sum_{s\in S} z_s \sum_{l\in s} \lambda_l + \max_{\boldsymbol{p}\geq 0} \sum_{f\in\mathcal{F}} p_f \sum_{l\in f} \lambda_l$$

$$\text{s.t.} \quad \sum_{f\in\mathcal{F}} p_f = 1.$$

(scheduling)

- An NP-hard combinatorial Max Weighted Independent Set problem

$$\max_{\boldsymbol{p}\geq 0} \quad \sum_{f\in\mathcal{F}} p_f \sum_{l\in f} \lambda_l \quad = \quad \max_{f\in\mathcal{F}} \sum_{l\in f} \lambda_l$$

$$\text{s.t.} \quad \sum_{f\in\mathcal{F}} p_f = 1.$$

Related Work on Scheduling

- Wireless scheduling is NP-hard [Lin-Shroff-Srikant 06, …]
- It is recently shown that bottom-up CSMA can solve the scheduling problem approximately
  - [Wang-Kar 05, Liew et. al. 08, Jiang-Walrand 08, Rajagopalan-Shah 08, Liu-Yi-Proutiere-Chiang-Poor 09, Ni-Srikant 09, …]
- Our framework provides a new top-down perspective
  - Note that our framework applies to general combinatorial problems

Step 1: Log-sum-exp Approximation

$$\max_{f \in \mathcal{F}} \sum_{l \in f} \lambda_l \approx \frac{1}{\beta} \log \left(\sum_{f \in \mathcal{F}} \exp \left(\beta \sum_{l \in f} \lambda_l \right) \right)$$

- Approximation gap: $\frac{1}{\beta} \log |\mathcal{F}|$
  - The approximation becomes exact as β approaches infinity

Step 1: Log-sum-exp Approximation

$$\max_{f \in \mathcal{F}} \sum_{l \in f} \lambda_l \approx \frac{1}{\beta} \log \left(\sum_{f \in \mathcal{F}} \exp \left(\beta \sum_{l \in f} \lambda_l \right) \right)$$

$\|$

$\|$ Log-sum-exp is a concave and closed function, double conjugate is itself

$$\max_{\boldsymbol{p} \geq 0} \quad \sum_{f \in \mathcal{F}} p_f \sum_{l \in f} \lambda_l$$
$$\text{s.t.} \quad \sum_{f \in \mathcal{F}} p_f = 1.$$

$$\max_{\boldsymbol{p} \geq 0} \quad \sum_{f \in \mathcal{F}} p_f \sum_{l \in f} \lambda_l - \frac{1}{\beta} \sum_{f \in \mathcal{F}} p_f \log p_f$$
$$\text{s.t.} \quad \sum_{f \in \mathcal{F}} p_f = 1.$$

Big Picture After Approximation

- The new primal problem

$$\max_{\boldsymbol{z}\geq 0, \boldsymbol{p}\geq 0} \quad \sum_{s\in S} U_s(z_s) - \frac{1}{\beta}\sum_{f\in\mathcal{F}} p_f \log p_f$$

$$\text{s.t.} \quad \sum_{s:l\in s, s\in R} z_s \leq \sum_{f:l\in f} p_f, \quad \forall l \in L$$

$$\sum_{f\in\mathcal{F}} p_f = 1.$$

Distributed?

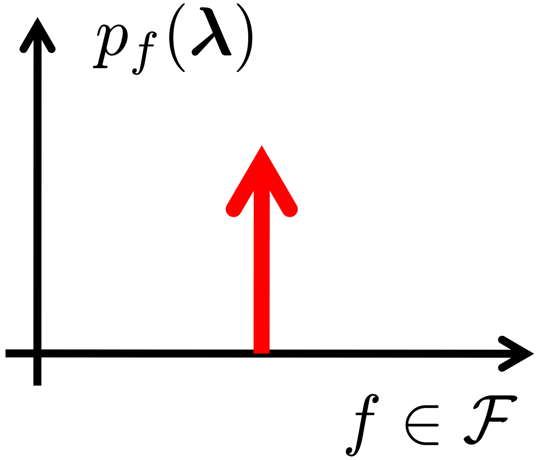
- Solution:

$$\begin{cases} \dot{z}_s = \alpha_s \left[U_s'(z_s) - \sum_{l\in s} \lambda_l\right]_{z_s}^{+} \\ \dot{\lambda}_l = k_l \left[\sum_{s:l\in s, s\in S} z_s - \sum_{l\in f} p_f(\beta\boldsymbol{\lambda})\right]_{\lambda_l}^{+} \\ \text{Schedule } f \text{ for } p_f(\beta\boldsymbol{\lambda}) \text{ percentage of time.} \end{cases}$$

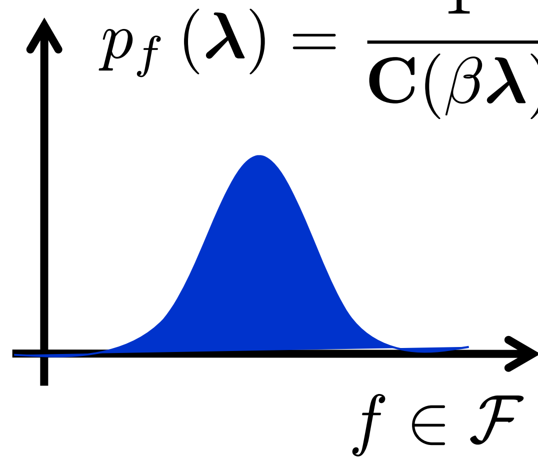
✔ ✔ ?

Schedule by a Product-form Distribution

$$\max_{f\in\mathcal{F}} \sum_{l\in f} \lambda_l \approx \frac{1}{\beta}\log\left(\sum_{f\in\mathcal{F}} \exp\left(\beta \sum_{l\in f}\lambda_l\right)\right)$$



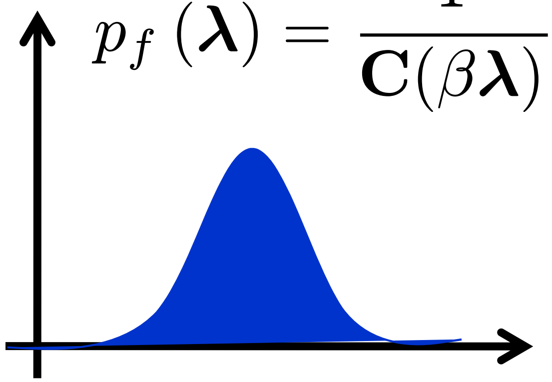
$$p_f(\boldsymbol{\lambda}) = \frac{1}{\mathbf{C}(\beta\boldsymbol{\lambda})} \exp\left(\beta \sum_{l\in f}\lambda_l\right)$$



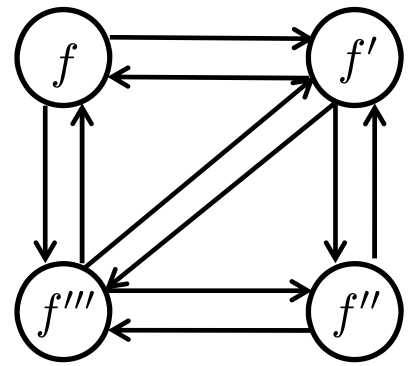
- Computed by solving the Karush-Kuhn-Tucker conditions to the entropy-approximated problem

Step 2: Achieving $p_f(\lambda)$ Distributedly

$$p_f(\boldsymbol{\lambda}) = \frac{1}{\mathbf{C}(\beta\boldsymbol{\lambda})} \exp\left(\beta \sum_{l \in f} \lambda_l\right)$$



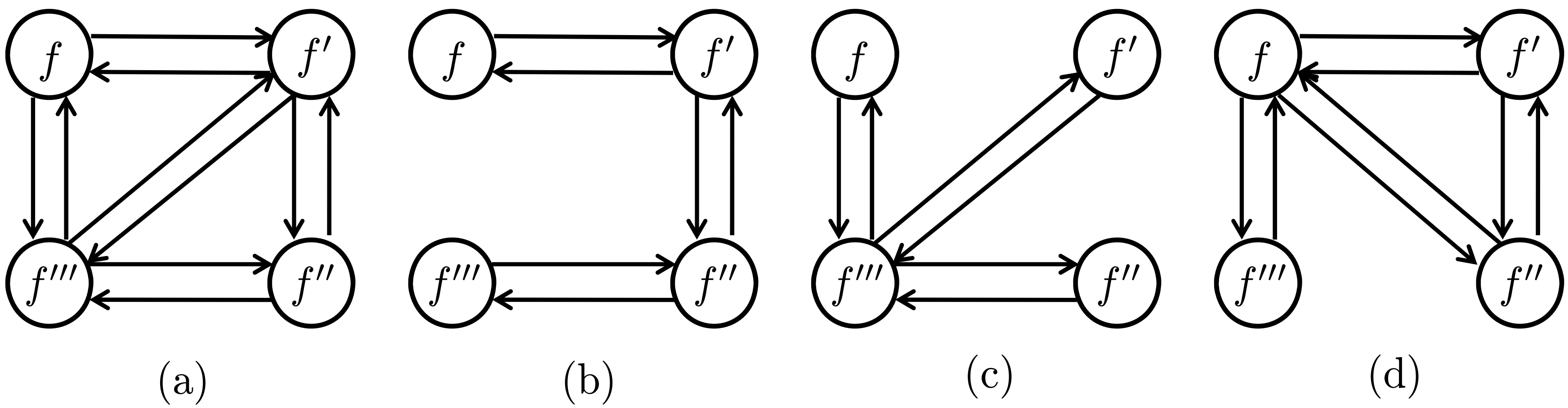
$f \in \mathcal{F}$



$$p_f(\boldsymbol{\lambda})\, q_{f,f'} = p_{f'}(\boldsymbol{\lambda})\, q_{f',f}$$

- Regard $p_f(\lambda)$ as the steady-state distribution of a class of *time-reversible* Markov Chains
  - States: all the independent sets $f \in \mathcal{F}$
  - Transition rate: **new design space**
  - Time-reversible: detailed balance equation holds

Design Space: Two Degrees of Freedom



$$p_f(\boldsymbol{\lambda})\, q_{f,f'} = p_{f'}(\boldsymbol{\lambda})\, q_{f',f}$$

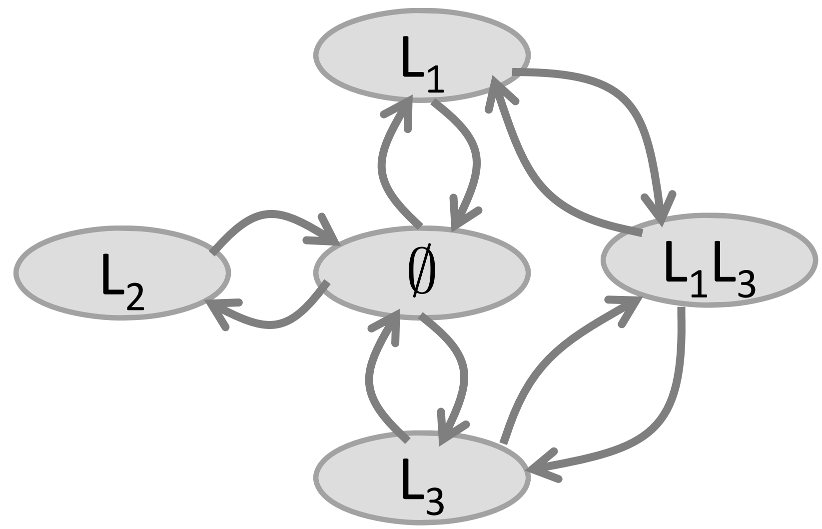
- 1) Add or remove transition edge pairs
  - Stay connected
  - Steady state distribution remains unchanged
- 2) Designing transition rate

Design Goal: Distributed Implementation

Implement a Markov chain

=

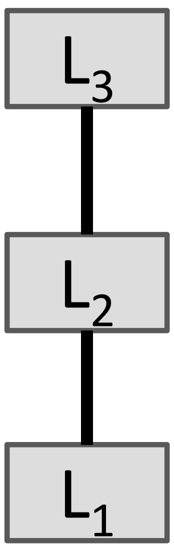
Realize the transitions



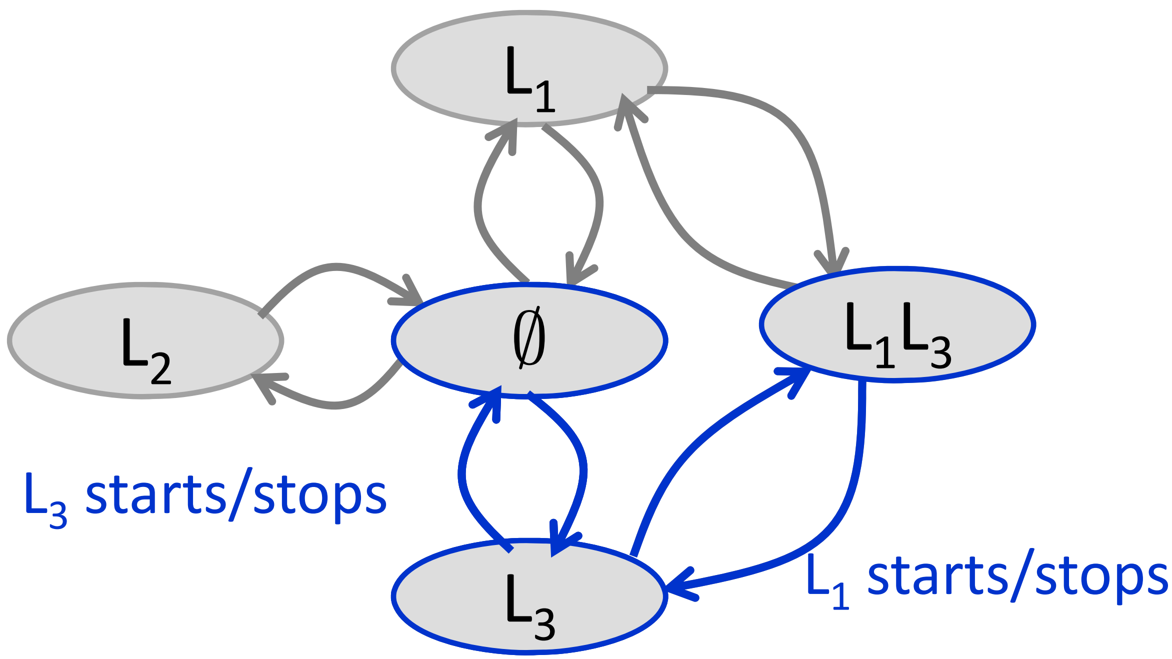
- What leads to distributed implementation?
  - Every transition involves only one link
  - Transition rates Involve only local Information

Every Transition Involves Only One Link

- From f to f' = f $\cup$ $\{L_i\}$: L_i starts to send
- From f' = f $\cup$ $\{L_i\}$ to f: L_i stops transmission



3-links conflict graph



Designed Markov chain

Transition Rates Involve Only **Local Information**

- Consider transition between f and f' = f $\cup$ {L_i}
- λ_{L_i} is available to L_i locally

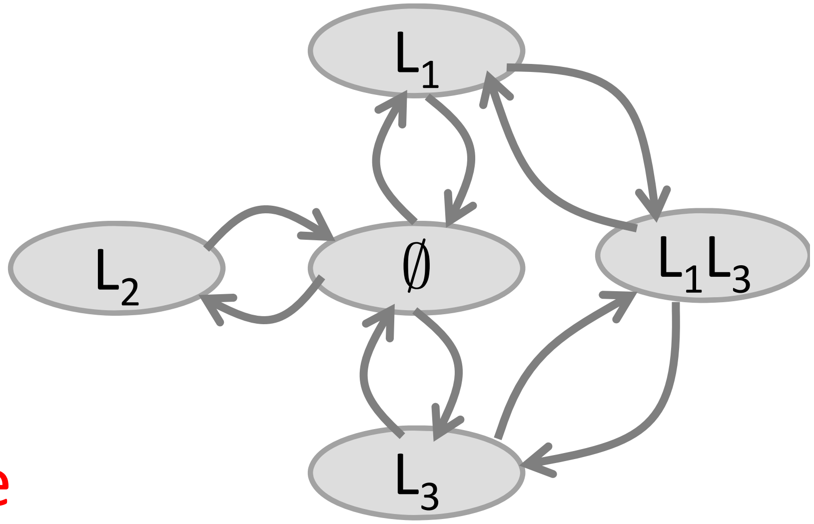
$$\frac{\exp\left(\beta \sum_{l \in f} \lambda_l\right)}{\mathbf{C}(\beta \boldsymbol{\lambda})} q_{f,f'} = \frac{\exp\left(\beta \sum_{l \in f'} \lambda_l\right)}{\mathbf{C}(\beta \boldsymbol{\lambda})} q_{f',f}$$

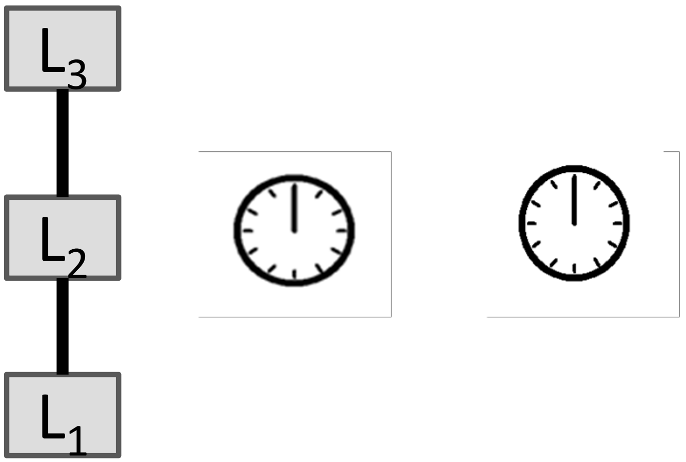
$$q_{f',f} = 1$$

$$q_{f,f'} = \exp\left(\sum_{l \in f'} \beta\lambda_l - \sum_{l \in f} \beta\lambda_l\right) = \exp \beta\lambda_{L_i}$$

Distributed Implementation

- Link L_i counts down at rate $\exp(\beta \lambda_{Li})$
  - Count down expires? transmit
  - Interference sensed? Freeze the count-down, and continue afterwards





- Reinvent CSMA using a top-down approach

The Total Solution

Distributed?

$$\begin{cases} \dot{z}_s = \alpha_s \left[U_s'(z_s) - \sum_{l \in s} \lambda_l \right]_{z_s}^{+} \\ \dot{\lambda}_l = k_l \left[\sum_{s: l \in s, s \in S} z_s - \sum_{l \in f} p_f(\beta \boldsymbol{\lambda}) \right]_{\lambda_l}^{+} \\ \text{Distributed MCMC achieves distribution } p_f(\beta \boldsymbol{\lambda}). \end{cases}$$

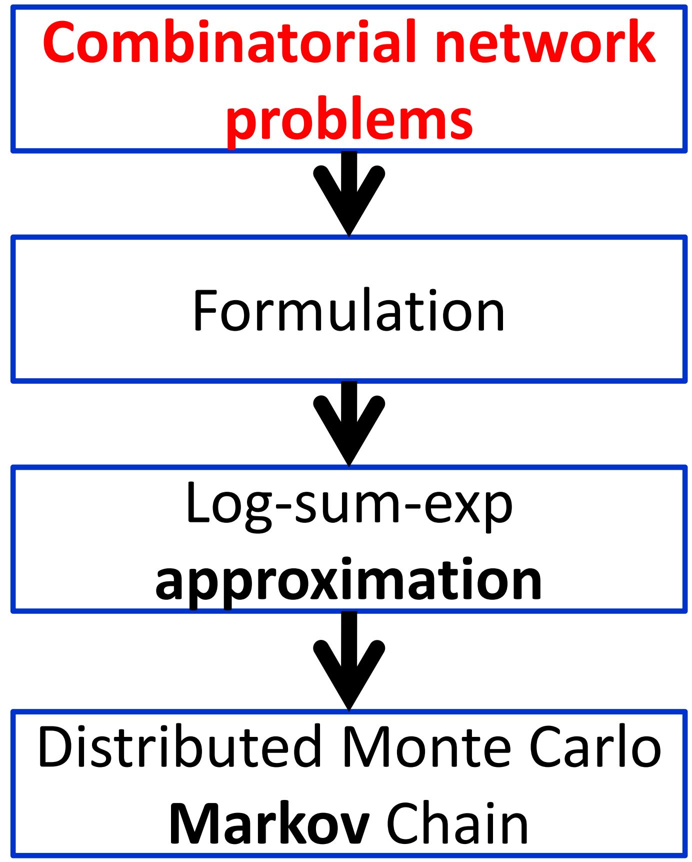
✓ ✓ ✓

- The distributed system converges to the optimal solution
  - Proof utilizes stochastic approximation and mixing time bounds

Conclusions and Future Work

Combinatorial problem

- Top-down approach
  - Combinatorial problems
  - **Markov approximation for designing distribution solutions**



□ Future: Convergence (mixing) time, and applications

Thank you

Minghua Chen (minghua@ie.cuhk.edu.hk)
http://www.ie.cuhk.edu.hk/~mhchen/